# *Atari Computer Enthusiasts [N.S.W.]*

| INSIDE INFO | No. 13 | June 1984 |
|---|---|---|
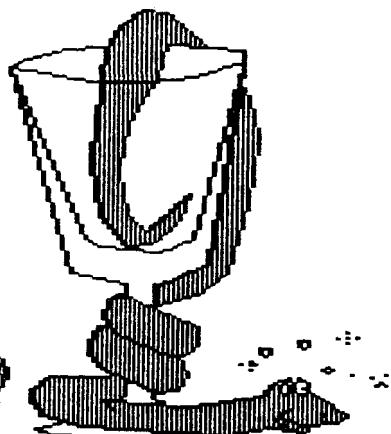
©1983
JBB

©1983
JBB
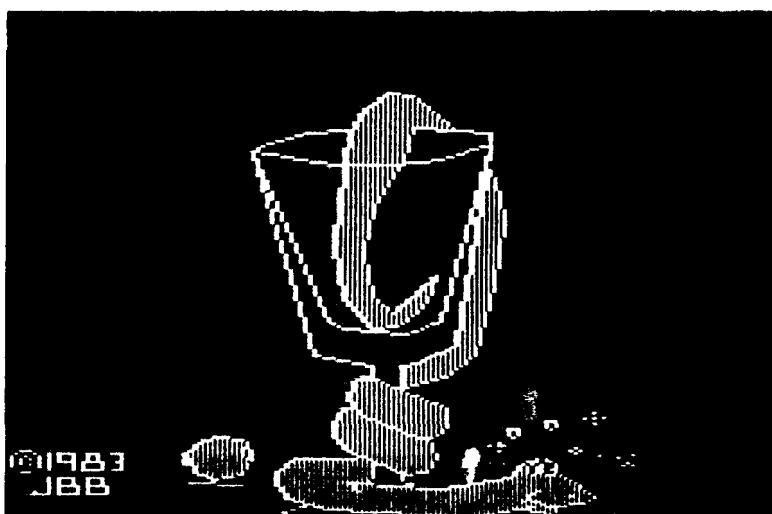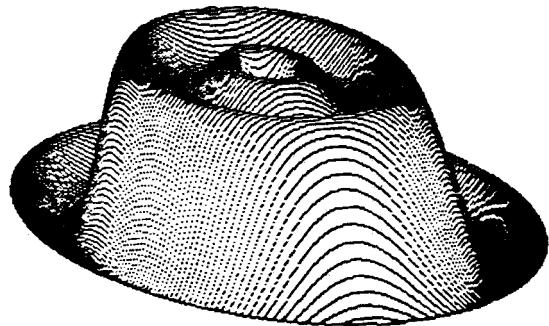
Atari Computer Enthusiasts (N.S.W.) is an independent, non-profit computer users' group loosely affiliated with Atari Computer Enthusiasts in the U.S.A. We have no connections with ATARI, Inc. or their Australian distributors, Futuretronics Australia Pty. Ltd. Our aims include promotion of the various ATARI Home Computer Systems, instructing both beginners and advanced users in programming techniques, exchanging public domain software, hints, tips and ideas amongst members and generally enjoying ourselves. The Club cannot condone software piracy.

| COMMITTEE MEMBERS | Home Phone # |
|---|---|
| Tony Reeve (President) | (02)452 2974 |
| Barry Williams (Vice President) | (02)452 2229 |
| Alex Kwok (Secretary/Treasurer) | (02)661 3377 |
| Peter Bamford (Meetings Organizer) | (043)42 2655 |
| Peter Dickeson (Cassette S/W Xchge) | (02) 80 2505 |
| Brian Simmons (Disk S/W Xchge) | (045)76 5331 |
| Chris Fitzgerald (Editor) | (02)637 5447 |
| Roberto Romano (Bulletin Board) | (02)327 4898 |
| Jeff Maddock (Librarian) | (02)568 2990 |

Address all mail to the relevent committee position:-

C/- Atari Computer Enthusiasts (N.S.W.)
G.P.O. Box 4514,
Sydney,
N.S.W. 2001.

## MEMBERSHIP FEES

$30 for the first year and $15 thereafter (or $20 and $10 respectively for students under 18 and still at school). Overseas air-mail subscriptions are $10(Aust) extra per year. Please write to the Secretary for further details.

## SOFTWARE EXCHANGE

Currently, there are 5 titles available to members, on either disk or tape. These are:

*GAMES #1    *UTILITIES #1    *fig-FORTH
*INSIDE INFO Vol. 1    *XL TRANSLATOR (disk only)
*INSIDE INFO Vol. 2 with CARTOONS

DISKS: $10 each.  CASSETTES: $8 each.

Non-members add 10%. Include an extra $1.50 to cover surface postage within Australia. Contact appropriate software exchange for further details. Overseas members should write for costs first.

For every program you write and have accepted by the Software Exchange, you will be entitled to one Software Exchange disk or tape at half price.

We are keen to exchange software with other user's groups.

## BLANK MEDIA

DISKS: $32          CASSETTES: $9

Prices are for boxes of 10 and may change without notice. An extra $1.50 should be included to cover postage within Australia. Contact appropriate Software exchange for details.

## DISK BOXES

These hold 90 disks and have a lockable lid. Cost to members is $35, non-members $39.50. Include an extra $3.35 for postage within N.S.W. and $5.50 elsewhere within Australia. Contact club Secretary for details.

## SPECIAL INTEREST GROUPS (S.I.G.s)

Home phone numbers of contacts are given below. Phone them or write to the Editor with your suggestions, high scores, reviews, hints or tips.

ADVENTURES: Contact Garry Francis    (02)789 1397
ARCADE GAMES: Contact Ken Shiu       (02)534 2120
COMPUTE!: Contact Barry Williams     (02)452 2229
FORTH: Contact John Mattes           (02) 94 5463
HARDWARE: Contact Jamie Athas        (02)349 7365

## A.C.E. BULLETIN BOARD -Phone(02)327 4898

Systems Operator (SYSOP) is Roberto Romano. System available between 8.30 P.M. and 9.30 P.M. Tuesday to Saturday only. Access requires a personal password which is available from SYSOP for $3 per year. We recommend you use "AMPLUS3.UDL" terminal software. PUBLIC DOMAIN SOFTWARE ONLY!

## INSIDE INFO

This is our bi-monthly users' group magazine. Articles, etc. should reach the editor at least one month ahead of the release date. Authors of articles printed in INSIDE INFO are entitled to one disk or cassette from the Software Exchange at half price.
BACK ISSUES: $2.50 each for issues in stock. $4 each if out of stock. Mailing costs are extra.
EXCHANGE SUBSCRIPTIONS: We are keen to exchange magazines with other Atari Users' Groups. Please write to the Secretary for details.
COPYRIGHT: Unless otherwise stated, articles in INSIDE INFO are not copyrighted. However, if any article is reprinted, acknowledgement of our source should be made. Also, please send TWO copies of the issue of your magazine to A.C.E. (N.S.W.).
ADVERTISING: Contact the Vice President for costs and conditions. Members' personal adds are free.

## REFERENCE LIBRARY

Some books are brought to meetings, others are available only by arrangement with the librarian. Loans are made only to people researching articles or talks for the club.

## DISCOUNTS AVAILABLE TO MEMBERS

Present your membership card for discounts!
COMPUTERWAVE [5% -cash only] 325 George St, Sydney.
(near Wynyard) Ph. (02) 29 1631.
THE COMPUTER SPOT [5% credit, 10% cash] Shop C4,
M.L.C. Centre, Martin Place. Ph.(02)235 2971.
COVER CARE [25% off computer dust covers] P.O. Box 719, Chatswood, N.S.W. 2067. Ph.(02)498 5631.

## MEETING DETAILS

Meetings are held at 6.00 P.M. on the first Monday of the month (or the second Monday if it clashes with a public holiday) in the Amenities Room, 7th Floor, OTC House, 32-36 Martin Place, Sydney.
NOTE: You can't enter the building after 6.30 P.M. and you MUST sign in!

## MEETING DATES FOR 1984

| | | |
|---|---|---|
| 2nd July | *13th August | 3rd September |
| *8th October | 5th November | *3rd December. |

## MEETING DATES FOR 1985

| | | |
|---|---|---|
| *4th February | 4th March | *1st April |

* indicates release dates for INSIDE INFO.

# SOFTWARE COMPETITION RESULTS

by Peter Dickeson. (Software Exchange)

The results of the Educational Software Competition were announced by Tony Reeve at the May meeting. There were only four entries, but the general standard of the entries was very high.

The members of the committee (software exchange) used the following criteria to examine the submissions:-

| | | |
|---|---|---|
| Educational value | 40% | |
| Documentation | 20% | |
| Ease of use | 20% | |
| Presentation | 10% | |
| Originality | 10% | |

Jamie Athas was unfortunate to meet such strong opposition. His program, titled "Basic Electricity", was cassette based and featured his voice to help explain the examples shown on the screen. He also asked questions which had multiple choice answers to further reinforce the lesson. The program was presented on an autoboot cassette and was very well protected.

The THIRD prize of $25 went to Peter Tucker of Lismore. He wrote a program titled "Spelling B" which was also cassette based. Like Jamie's program, "Spelling B" made use of the left track of the cassette tape.

Your Atari only uses the right channel of the tape, leaving the left channel free for voice, music, noise or all three. Peter's voice track described the program as it loaded as well as "saying" the word which was to be spelt. Each word was accompanied with an explanation or description of what the word meant -this would be played through the television speaker.

"Spelling B" allows you to use your own words, giving it great flexibility and making it an excellent program for small children who are just beginning to spell.

[EDITOR'S NOTE: An article on using your program recorders left track has been prepared by Peter and can be found elsewhere in this issue of Inside Info.]

"Graphs4", by Bao Quy Nguyen Phuoc, won the second prize of $50. It is one of the most flexible and easy to use 'graph plotter' programs that I have ever seen.

There are obvious applications for this program in high school Mathematics and Physics. However, any subject that makes use of graphs, e.g., Economics, Geography, etc., would benefit. Up to three functions of y may be plotted with the user defining the origin and unit length of each axis.

Bao made use of the Ataris "forced read mode" so that your formula could be input in the same way as it would be written into any BASIC program.

Ken Shiu won the first prize of $100 for his exceptional program "QB Lightfingers". The program is a very effective touch typing course in the form of a game.

Provision has been made for anyone from beginner to expert. There are levels of play within each section of the program with the speed being automatically incremented upon successfully finishing a section. This all adds up to make for a very sophisticated "game".

Graphics and sound are excellent while the supporting documentation is the most complete that I have ever seen.

Congratulations to all who submitted their programs, I'm sure that the well earned prize money will come in very handy.

# FOR SALE

# ADD AUDIO TO YOUR PROGRAMS

by Peter Tucker. (Lismore, N.S.W.)

[Editor's note:- Peter won the Third Prize of $25 in the clubs' Educational Software Competition with a program he wrote called "SPELLING B". This program made use of the sound track capabilities of the Atari program recorder. This article describes how he recorded that soundtrack and should be useful for others wanting to do the same.
CAUTION: While the method outlined in this article does work, we cannot accept responsibility if you do any damage to you program recorder. Tampering with your recorder will invalidate any 90 day warranty on the unit!]

As most Atari owners are aware, the 400/1010 program recorders have the ability to replay audio as well as digital information. This is achieved by the recorders' having a two channel (stereo) playback head.

The digital information is stored as audio tones (5327Hz=logic 1 and 3995Hz=logic 0) on the right (inner) track while the audio (sound) is recorded on the left (outer) track.

|          | AUDIO TRACK   | LEFT TRACK  |
|----------|---------------|-------------|
| SIDE A   |               |             |
|          | DIGITAL TRACK | RIGHT TRACK |
|          | AUDIO TRACK   | RIGHT TRACK |
| SIDE B   |               |             |
|          | DIGITAL TRACK | LEFT TRACK  |

Unfortunately, Atari made no provision for recording audio information, however, this may be successfully accomplished using a conventional recorder.

There are two main applications for the audio track:-

1. Giving audio information while the program is loading. e.g. directions on how to use the program.

2. Having the program access the recorder for varying forms of information. e.g. Atari's language programs.

## METHOD 1

Setting up cassettes for the first application, whilst being the easiest to record (since there are no serious timing limitations), requires special hardware.

Conventional recorders will erase the digital information as the audio track is being recorded. If you have the ability to record sound on sound (rare in cassette decks) then it is a simple matter to record audio on the left track. For those of us who do not have this facility there are still ways left open to us .

It would be possible, though I have not personally tried it, to transfer the digital track to a reel to reel tape with dubbing facility, lay down the audio track and then transfer it back to cassette. This method has obvious limitations:-
a) finding a suitable reel to reel machine.
b) the digital information would suffer a reduction in quality due to the double audio copy.

I have experimented with disabling the erase head in a conventional stereo recorder and have had moderate success. Unfortunately, while the digital track is not erased, the record head affects the recording even though no microphone is connected to the right channels input jack. The result of this is that the volume of the digital recording is severely reduced. Tapes made in this manner will load but, with my experience of the long term volatility of cassette recordings, I would not recommend this method.

## MODIFYING YOUR RECORDER

By far the most successful method is to lay down the digital track as the last step. This is achieved by 'attacking' the program recorder itself. The erase head can be easily disabled using the following method. (My recorder is an older 410 -no pause control with the tape counter situated on the right, near the front of the cassette bay. Hence my directions may vary slightly for the later model 410s and the 1010).

1. Remove the two screws from the back of the recorder.
2. Gently lift the back off the recorder starting from the control button end of the machine. The other end (with the serial cable outlet) slips under two small lugs.
3. Loosen the philips head screws which retain the cassette bay lid pivots, remove the spring and release the lid from the body of the recorder.
4. Undo the screws which retain the component board and motor mechanism to the plastic casing (do not forget the screw near the counter mechanism).
5. Gently remove the internal unit from the plastic case.
6. Turn the internal unit over. It is now a simple matter to remove the small philips head screws which retain the erase head.
7. Retract the head, complete with wires, and tape it safely out of the way. Take care that it will not get in the way of the cassette head mechanism or foul when reassembling the unit.
8. Reassemble the recorder.

The above instructions may sound complicated but it is, in reality, very simple.

## MAKING THE TAPE

1. Load your program into the computer or keep a back-up copy of the program since it will be erased from the tape.
2. Place the cassette in a conventional stereo recorder and lay down the required audio track.
3. Place the cassette back in the program recorder and save the program back to tape.
You should now have a perfect copy of your program complete with audio track.

Apparently, the 410 program recorders recording circuitry is single track, since the audio track

suffers no loss of volume or quality while saving the program (digital) track. It would however, be no problem if it did drop in quality, since you could always compensate by turning up the TVs volume control.

Don't forget to replace the recorders erase head when finished because even though programs will successfully load from your recorder without it, future saves to pre-recorded tapes will be garbled.

A leader program should be used to turn off the usual program LOADing sounds, otherwise these sounds will be heard along with your specially recorded audio track. This can be achieved by POKEing location SOUNDR (Decimal 65) with a zero. i.e.:-
                    POKE 65,0
This leader program could also be used to chain the main program, e.g.:-

    10 POKE 65,0:REM TURN OFF SOUND
    20 REM PERHAPS SOME SORT OF SCREEN DISPLAY WITH
    30 REM INSTRUCTIONS COULD GO HERE
    40 CLOAD:REM CHAIN MAIN PROGRAM

## METHOD 2

Where you only want to use the program recorder to play back audio information during the actual RUNning of the program, you don't need to remove the erase head since there will be no information recorded on the digital (Right) track. These programs require the program recorder to be turned on and off so that the sound track can be heard before the program continues RUNning. This requires a short program routine which will control the program recorders motor as well as the period of time that the motor will run for. Motor control can be achieved by POKEing PACTL (Decimal 54018) with 52 to turn it on and 60 to turn it off. i.e.:-

    10 POKE 54018,52:REM TURN RECORDER ON
    20 REM TIME DELAY WHILE RECORDER PLAYS
    30 POKE 54018,60:REM TURN RECORDER OFF

The timing delay can be controlled with a simple FOR/NEXT loop, however the actual time for any given loop will depend on a number of factors. FOR/NEXT loops near the end of a program (especially a long program) will execute more slowly than the same loop placed at the start of a program. This is because the 'NEXT' command in BASIC always jumps back to the first line number of the program when it is searching for its matching 'FOR' statement. The current GRAPHICS mode will also affect execution time.
Providing that line numbers within a program do not change and that the GRAPHICS mode doesn't change, the period of the timing loop will not change.

A much cleaner method is to use one of the in-built system timers, of which there are a number built into the Atari. e.g. Decimal 540, 542, 18, 19, & 20. Timers at memory location 540 & 542 count backwards from 255 to 0, whilst memory locations 18, 19 & 20 (which are most suited for our purpose) count and increment from 0 to 255.

Location 20 counts from 0 to 255, being incremented by 1 every 1/50th of a second (for a 50Hz

power supply). Hence it takes 5.12 seconds for location 20 to reach 255. Location 20 then zeroes again and location 19 is incremented by 1. When location 19 is incremented to 255 (every 1310.72 seconds, or 21 minutes 50.7 seconds), it zeroes and increments location 18 by 1.
In order to get accurate timing using these memory locations, they should all be zeroed (POKEd with 0) just prior to use. Here's a sample program:-

    10 POKE 20,0:POKE 19,0:POKE 54018,52:REM zero timers, turn cassette on
    20 IF PEEK(19)<2 THEN 20
    30 POKE 54018,60:REM turn cassette off
In line 20 the program waits for location 19 to be incremented to two or more, hence the program will wait approximately 10.24 seconds (i.e. 2*5.12) whilst incrementing 0 to 1 then 1 to 2.

## RECORDING THE SOUND TRACK

Once the time interval has been decided upon and inserted into your BASIC program, you can move on to recording the sound track. After much experimentation I have found the following method to be both simple and effective:-
1.    Save your leader and BASIC programs to tape -do not rewind the tape!
2.    Obtain a stopwatch or watch with seconds clearly marked on it.
3.    Place the tape which is to carry the sound track into the conventional stereo recorder.
4.    Zero the tape counter at the point where the BASIC program ended.
5.    With the microphone connected to the RIGHT track (channel) only, press PAUSE and set machine for recording.
6.    Release the PAUSE key and simultaneously start your watch.
7.    Record a short, sharp sound every n seconds (n being your predetermined time interval) -I have found that tapping the side of a glass with a spoon is very suitable.
8.    Record as many intervals onto the tape as you have pieces of information to be played.
9.    When your timing marks are all recorded, rewind the tape back to the zero position.
10    Connect the microphone into the LEFT track (channel) and press PAUSE.
11    Set machine for recording.
12    Release the PAUSE key and record your sound track. Press STOP before you exceed your predetermined time (n).
NOTE: It is wise to, if possible, record your audio as close as possible to the centre of the timed interval -this will make it less critical if you can't position your tape accurately to the start of your "timing mark".
13    Press PLAY and listen for the short, sharp sound that you recorded in 7 (above) then press PAUSE again.
14    Repeat steps 11 to 13 for as many intervals as you require.

Your tape is now completed. Rewind it and place it into your program recorder.                        ☆☆☆

**5**

# WIRING AN EXTRA KEYBOARD

by Peter Banford, (Umina, N.S.W.)

If you own a 400 and you're fed up with its membrane keyboard or if you just want a seperate keyboard for your 800 and you've got around $50.00 to spare, then this may be for you -a seperate full stroke keyboard. To build the keyboard you will need the following:-

1) Keyboard -I used a MicroBee keyboard from Jaycar Electronics at $29.95.

2) Extra Key(s) -$1.50 from Jaycar.

3) Ribbon Cable -(at least 22-conductor) about $1.00 per metre.

4) 25 pin male & female 'D' Type connectors with covers at approx $4.50 each.

All of the above components are readily available- check the electronics magazines for the best prices.

## EXTRA KEYS

Some of the keycaps on the MicroBee keyboard are different to the ATARI keys, so I used model paint to paint over them. I then used a fine pen to draw the correct symbols. If you press hard enough the pen cuts into the paint and the symbols will last longer.

The MicroBee keyboard only comes with 60 keys whereas you need 61 for the Atari (including the Console keys). It's also a good idea to use 2 keys in series for SYSTEM RESET to avoid accidental use. For the other console keys I used 3 large keys (i.e. 'RETURN' size) -hence the need to buy extra keys. The keyboard already has the extra holes for up to 6 additional keys so there's no problem there.

## THE PROCEDURE

The wiring diagram in Figure 1 originally came from an article in Creative Computing (Dec. 1982) by Robert Noskowicz. It shows the standard Atari keyboard from the back -the side you'll do the soldering from.

1 First of all, wire-up the back of the keyboard as shown.

### HINTS

1) Tin all wires and key terminals before actually soldering.

2) Follow the diagram closely, connecting wires in proper sequence i.e. all those connected to 1, then 2, etc.

2 Next, connect the ribbon cable to the keyboard ensuring that you keep the correct sequence.

3 If you are using the 'D' connectors, wire the socket (female) to the ribbon cable which is now connected to your external keyboard -take care that you wire it correctly. i.e. The 'D' connectors pins are numbered (and marked) from 1 to 25, ensure that the wires (numbers on the arrows in Figure 1) are connected to that same pin number.

If you want to permanently wire the keyboard to the computer you won't need the 'D' connectors -but this means your computer won't be as portable, nor will you be able to add a seperate numeric keypad later.

4 To mount the 'D' plug (male) in the computer you will need to drill a hole in the side. There's plenty of room and if you take your time and carefully file the hole to size a very neat fit can be achieved. Don't fix the plug in place yet though! [Editor's Note:- I put my connection in the back, but it is a very neat fit and extreme care should be taken.]

5 The connection to the computer is comparatively easy.

On the 400, remove the top half of the computer housing and solder the ribbon cable to where the existing cable for the keyboard connects. NOTE: No.1 corresponds to No.1 on the Atari keyboard cable (yep, if you look hard it is numbered on the PC Board).

For 800 owners the Console keys are not on the keyboard PC but there is a small connector (No. J202) on the main board to which you can connect them -just follow the PC tracks to check which switch (key) it operates from. No.22 is an earth and it can be connected to any common earth on the board.

[Editor's Note: I've only worked on my 800, but I would suggest you follow the existing wiring colour code for the main keyboard which I've set out in diagram 3.

NOTE: Point 18 (Grey) is not connected on the 800.]

6 Once again, if you are using the 'D' connectors, solder the plug (male) to the ribbon cable taking care to wire it correctly. i.e. ensuring that the wires (numbered 1 to 17 from right to left on the 800) are connected to that same pin number.

7 Now test the original Atari keyboard, if it works, plug in your new keyboard and try it. If it doesn't then check all the connections for bad joints or short circuits. Also, check to see you have No.1 to No.1 etc.

8 Either bolt or epoxy the 'D' plug (male) into the body of the computer and reassemble it.

The new keyboard is in parallel with the Atari keyboard, so both keyboards can be used simultaneously. (It's great for two player games that use the keyboard.)

To mount the keyboard I used plywood and spare bits of timber to make a case. I then covered it with self adhesive wallpaper (refer fig.2).

That's it, it takes a while with a lot of soldering but it's a lot cheaper than the commercial alternatives.

If you have problems I may be able help - just ask.

Incidently, you use the same method to fit a numeric keypad, but that's another article perhaps. [Editor's Note: I've connected up my own numeric keypad and I hope to put an article on wiring one up in the next Inside Info. In the meantime, if you want to buy one, they're available from Dick Smiths Electronics for $19.95. They come on a PC Board and hence require a little modification.]
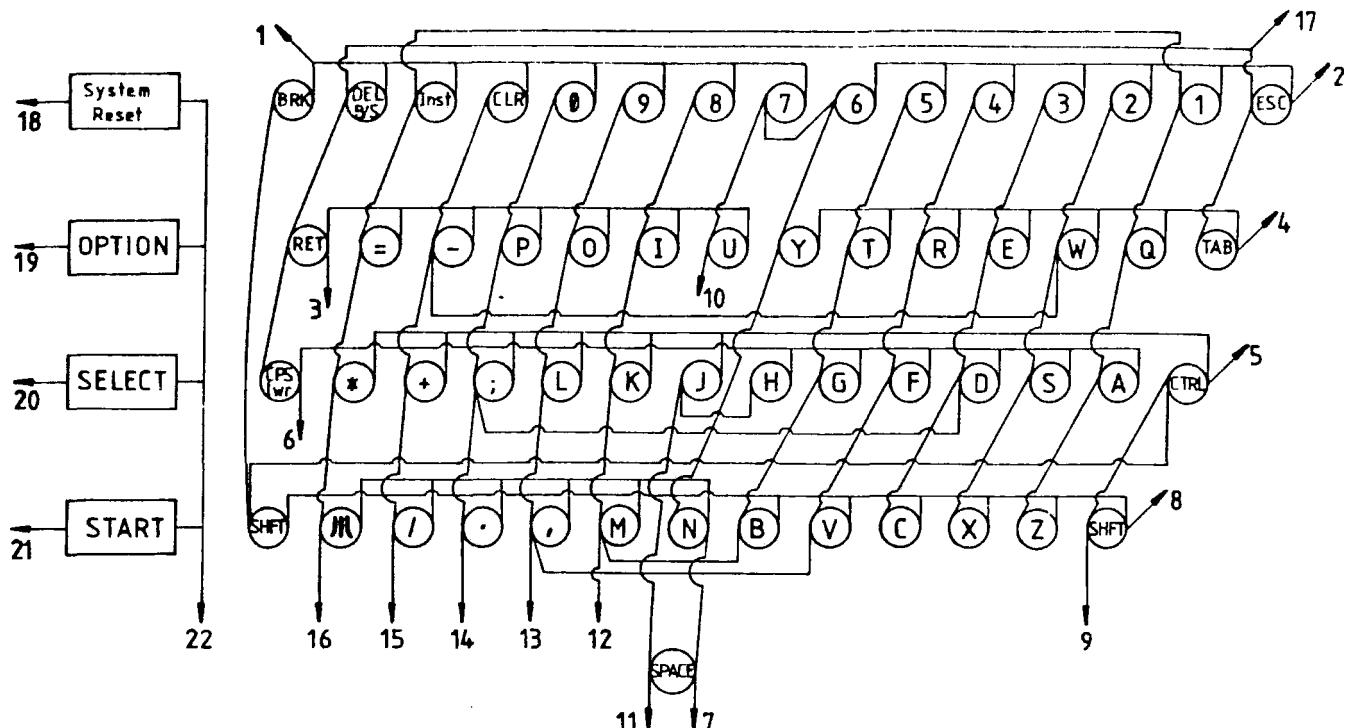
**Figure 1**
Keyboard Wiring Diagram
(viewed from back, front)
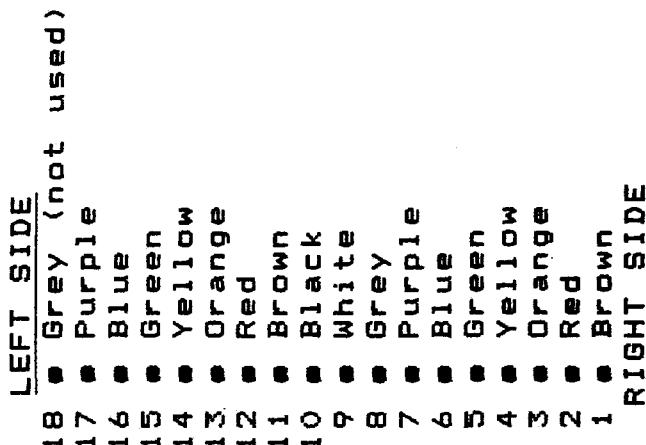


**Figure 3**
PC Board connection - underside of 800
(viewed from front)



Cut hole for keyboard

3 ply

Timber - any size
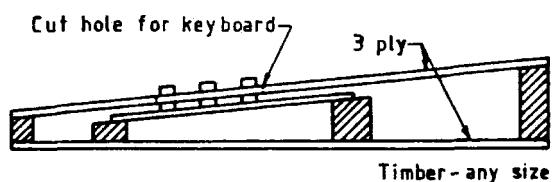
**Figure 2**
Keyboard Housing



# COMPUTE! PROGRAM EXCHANGE

From Barry Williams.  Phone (02)452-2229.  (Frenchs Forest, N.S.W.)

*******************************************************************

The exchange is now operating.  Programs will be supplied (swap one for one) with either 4 or 8 on a disk. Please send your exchange disk with four  or eight 'COMPUTE' programs on it and list which ones you would like in return.  You must also send verification that  you have the corresponding copy of 'COMPUTE'.  This can be done by sending a business reply tag from the pages of the relevent issue.

Refer to page 18 for the current library list.

*******************************************************************

# LIGHTPEN PROJECT

by John Mattes. (Seaforth, N.S.W.)

This article describes how to construct a high accuracy light pen at a cost of about $20.

## PARTS LIST

You will need the following:-

1    A "Stabilo Boss" highlighting pen ($2.30)

2    A two microsecond phototransistor - a Motorola MRD300 from VSI Electronics, St Leonards is suitable ($3.18). This item is CRITICAL.

3    An LM311 comparitor ($1.40) and 8 pin DIL (dual-in-line) socket ($0.90).

4    One 100 ohm half watt resistor ($0.05).

5    A nine pin 'D' type female connector ($2.95) and cover ($3.50). Note that some covers won't fit under the lip of your computer, so be careful!

6    A length of flexible cable with at least 4 conductors ($2.75).

7    A miniature PCB (printed circuit board) mounting momentary action switch. This type of switch turns off when released ($0.65).

8    A small piece of "vero-board", which is insulated board with a matrix of holes drilled at 0.1 inch (yes INCH) centres, overlaid with parallel copper tracks ($2.85).

## CONSTRUCTION

After you have obtained your components:-

1 Remove the innards of the highlighting pen by inserting a small screwdriver between the black "business end" of the pen and the case, and lever the case up, a bit at a time, until the glue breaks.

2 Cut the vero-board to fit snugly inside the pen -you can scale off diagram 1. Note that the tracks on the vero-board run parallel with the axis of the pen. 3 Using a small drill (about 3mm) or a hobby knife, cut the tracks where an 'X' appears on diagram 1. There are seven points at which the tracks MUST be cut.

4 Drill or file out the ridges on the inside of the hole in the tip of the pen so that it will clear the flange on the base of the MRD300. A 6mm drill is suitable. 5 File off or bend back the locating tag on the base of the MRD300 so that it will fit into the tip of the pen.

6 Drill a hole through the closed end of the pen to suit your cable.

That's the hard part over! Now:-

7 Insert and solder the components where shown on the vero-board. Note that the view shown in Diagram 1 is the track side of the board. All of the components are inserted from the other side.

8 Pin 1 of the 8 pin DIL socket is usually marked with a small coloured arrow.

9 Use a battery and lamp or a multimeter to check that operation of the switch has the effect of connecting together the tracks on which the switch is mounted.

10 Make sure that the phototransistor is connected according to the inset in diagram 2. Note that the view of the phototransistor is "looking down the leads".

11 Insert the cable through the case and solder the four wires to the tracks where shown.

12 Secure the cable to the board to prevent the soldered connections from breaking when in use.

13 Insert the LM311 comparitor in its base. You may need to (carefully) bend the pins in slightly. When viewed from above, the number 1 pin on the LM311 will be marked with a dot or the 1-8 end will be identified with a U-shaped notch. If you need to, remove the LM311 with a small screwdriver, prizing each end up a little at a time, else disaster!

14 Push the wiring board into the case and replace the end cap. You will have to cut the case away to clear the switch, and you may have to trim the end cap slightly. Make sure the switch operates without binding on the case.

15 Finally, connect the other end of the cord to the pins on the 'D' connector (put the cover on the cord first!). Pin numbers are marked on the connector although you may need a magnifying glass to see them. Point P1 on the diagrams connects to pin 1 on the 'D' connector and so on.

You can test the pen using the BASIC listing given. Diagrams 1 and 2 were done with the pen, but with a lot more software. You can get this software (written in FORTH of course) through the software exchange. If you can't get it to work, ring me on (02)94-5463(AH). If all else fails, I will fix it for $20 plus components. Alternatively (given enough interest) I could make assembled and tested pens available to club members for $40.

As a final note, accuracy can be increased (at the expense of sensitivity) by reducing the 100 ohm resistor to, say, 68 ohms.

[Editor's Note: The uninitiated may have problems with reading the diagrams. Diagram 1 is supported by the circuit diagram in diagram 2 e.g. they both show that pin 8 of the LM311 is linked to the track labelled P7 (pin 7 of the 'D' connector, track P8 is linked to pins 1, 2 & 4 etc.]

```
1 REM ###################################
2 REM #          LIGHT PEN            #
3 REM #        by John Mattes         #
4 REM # Published by Atari Computer   #
5 REM #    Enthusiasts (N.S.W.)       #
6 REM #          June 1984            #
7 REM ###################################

10 GRAPHICS 7+16
20 SETCOLOR 4,0,14:COLOR 3
30 H=PEEK(564):H=H-74
40 IF H<0 OR H>159 THEN GOTO 30
50 V=PEEK(565):V=V-17
60 IF V<0 OR V>95 THEN GOTO 30
70 IF STICK(0)<>14 THEN GOTO 30

80 PLOT H,V
90 GOTO 30
```

0.1 inch spacing

DIAGRAM 1

MRD310  +5V  P7

B    C
     E    3  8 LM311
              7  P6
100         4
ohm     2      P1
        1   SWITCH
               P8

B
E ( . . ) C
MRD310
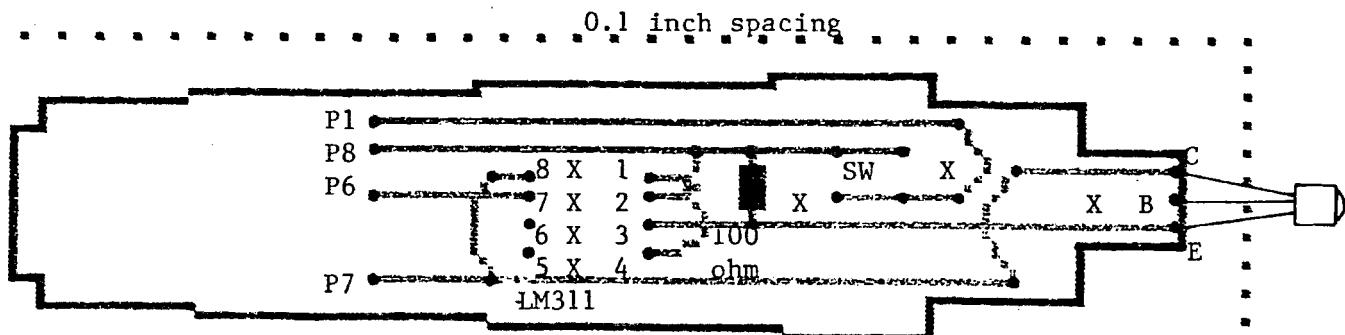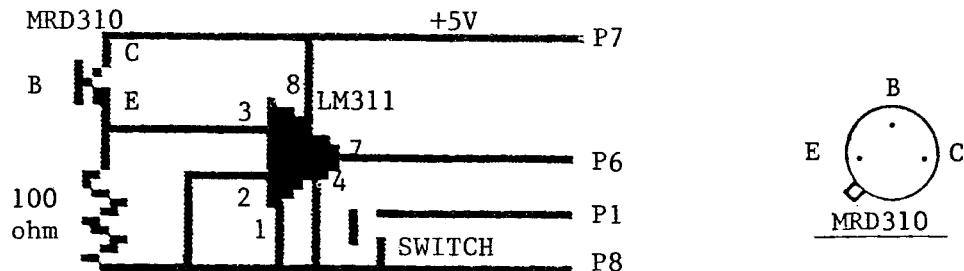
DIAGRAM 2

# OOPS!

Since I've taken over as editor I have made a few blunders with information that I've passed on through the pages of Inside Info. While the mistakes are generally fairly minor, I'd like to offer the corrections. You should go back through the issues mentioned and mark the corrections in -this will ensure that you don't refer back to incorrect information at a later date.

## INSIDE INFO #11

### LINEDITA PROGRAM Page 17

I found a bug in this program after Inside Info had been printed. I found that, if during the EDIT mode, you had pressed the "CAPS-LOWR" key or ATARI (inverse) key the program would not recognize your keystrokes when you went to enter a filename before SAVEing to disk -i.e. You would be typing in inverse or lower case characters which the program does not allow. The fix for this is to amend line number 32620 as follows:-

        32620 POKE 694,0:POKE 702,64:A=PEEK(Y):IF A=39 OR
                A=60 OR A=Z THEN 32620

In the event that you also renumbered the progam as stated to give LINEDITB.LST you will need to amend line number 70 as follows:-

    70 POKE 694,0:POKE 702,64:A=PEEK(Y):IF A=39 OR A=60
                OR A=Z THEN 70

## INSIDE INFO #12

### RECOMMENDED TERMINAL SOFTWARE Pages 2 & 5

I wrongly stated that the recommended terminal software for use with our own A.C.E. Bulletin Board was "JTERM". The correct terminal program is "AMPLUS3.UDL". This can be downloaded from the BBS using MINATERM.NSW when you make a connection.

### MINATERM.NSW PROGRAM Page 6

Whilst the program works as intended, when I first used it on the BBS I found it awkward because I couldn't see what I was typing on my screen. Roberto gave me this simple 'fix' which made it much easier to use. You can amend line 200 to read as follows:-
    200 IF PEEK(764)<255 THEN GET #1,C:? CHR$(C);:PUT
                #2,C:GOTO 190

### TALKING TO FUTURETRONICS Page 17

I stated in this article that Ataris are now manufactured in Taiwan, Futuretronics has advised us that this was incorrect. The new machines are manufactured in HONG KONG.

### APX POSTCODE Page 18

The postcode shown has a typographical error. It should be 95051.

### MIDEASTERN SOFTWARE Page 18

Since the magazine was printed MidEastern Software has changed it's phone number. The new number is:-
            0011-1-814-446-5117

### SPELLING ERRORS

Ok, ok, I'm not a yank! Just because I was up late, tired, hungry, exhausted and close to death. I'll try to make sure I don't use 'check' instead of 'cheque' in future. I'll also use 'led' instead of 'lead' and 'separate' instead of ''seperate'. All I can say is OOPS!

PS. It's ODDS & SODS not OODS & SOOS!!!!!!!

# A.C.E. MOVIE DIRECTOR

## PART 1

(or Page flipping, the easy way)
by Roberto Romano. (Bellevue Hill, N.S.W.)
NOTE: This article is the first one to reach our editor through the phone line.

This month I would like to unveil a powerful but hidden feature of the ATARI. I recently bumped into it while 'fooling around' with that marvellous machine. There are many hidden treasures to be found only by venturing "off the beaten track" of your BASIC textbooks.

Some of you perceptive programmers might have noticed that adding 32 to a GRAPHICS call inhibits the screen clear for that mode, in other words, after a program has run, it is still possible to view a graphics (or text) screen by calling the same Graphics mode +32. I'll explain this step by step and show you how it can be used to advantage.
We start off with a two liner:-

```
10 GR.24:C.1:SE.2,0,0:F.I=0TO317
STEP3:PL.160,96:DR.I,0:DR.I+1,191:DR.I+2,0:N.I
20 G.20
```

This gives us a simple drawing in GRAPHICS 8.
Just to get a simple message and a different GRAPHICS mode, type in these two lines:-

```
20 GR.18:POS.8,4:?#6;"HELLO THERE"
30 G.30
```

O.K. so far so good. Now comes the trick; add the following lines.

```
30 F.I=1TO300:N.I
40 GR.24+32:SE.2,0,0
50 F.I=1TO300:N.I
60 GR.18+32:G.30
```

Did you ever think page flipping was that easy?
Again, the lucid hackers will have noticed that this technique is not (yet) perfect. NOT YET, I said! If we look at the GRAPHICS 24 screen carefully, we notice a little "garbage" at the bottom. That garbage is simply the Graphics 2 screen (or whatever mode you might have chosen for the second screen) highly compressed. Of course the second screen must be of smaller memory requirement in order to see this effect.

Let's try to look at this a bit more technically. When you call a GRAPHICS mode, Antic puts the screen where it perceives the top of memory to be -e.g. if you call GRAPHICS 24 then approximately 8000 bytes (from the top of memory) will be reserved for its display. The TOP left hand corner of the screen corresponds to the BOTTOM of screen memory. So memory will fill up from low to high while the screen fills from top to bottom (for the purists...this is only a representation of what happens).

RAMTOP    <-
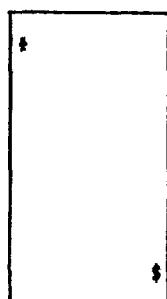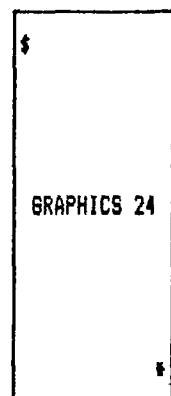


Memory          TV screen

In this diagram, the asterisks represent the last byte of screen memory and the dollar signs represent the first byte.
When, after having called GRAPHICS 24, you then call GRAPHICS 18, ANTIC first looks at the top of memory and starts filling the memory below it with the Graphics 18 stuff, eating its way through the GRAPHICS 24 screen. In this case, the amount lost is 12x20 or 240 bytes (12 lines x 20 bytes per line).

Try changing lines 20 and 60 in the above program for other graphics modes which use up more memory. You will see how much more space they eat out of the Graphics 24 display. GRAPHICS 7 should use up the bottom half.

After all this discussion, how do we get rid of this unwanted garbage at the bottom of the screen? The solution lies in reserving enough memory for each screen we want.

Knowing that ANTIC will start the screen memory where it thinks the top of memory is, all we have to do is to POKE location 106 (the top of memory - RAMTOP) with different values corresponding to where our seperate screens will be.

In our example, we need 32 pages for GRAPHICS 24 and 4 pages for GRAPHICS 18 (Remember that the display list cannot cross a 1K boundary without a JUMP instruction - that's 4 pages!). The minimum requirement for any screen is 4 pages - or 1k.
NOTE: "Pages" are 256 byte blocks of memory. 1k=1024 bytes (256*4). Hence a 48k system has 192 pages (48*4).

Theoretically then, we could have as many as 32 screens or "frames" in a 48K system,... allowing for the memory used by DOS and the BASIC cartridge and if your program is short enough. Short indeed it can be, as we shall soon see.

Let's now get a clean GRAPHICS 24 screen! Type in the following lines and RUN the program.
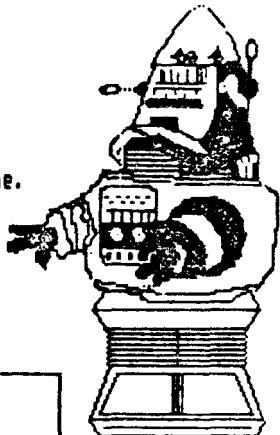
```
10 A=PEEK(106):DELAY=300
20 GR.24:C.1:SE.2,0,0:F.I=0TO319
STEP3:PL.0,0:DR.I,191:N.I
30 POKE106,A-32
40 GR.18:POS.8,4:?#6;"HELLO THERE"
100 GOS.DELAY
110 POKE106,A:GR.56:SE.2,0,0
120 GOS.DELAY
130 POKE106,A-32:GR.50
200 G.100
300 F.I=1 TO 300:N.I:RET.
```

So there you have it, two perfectly clean screens!
Let's now get a bit more involved and do 4 screens all of different GRAPHICS modes and let's "flip" through them. Add the following lines:

```
50 POKE106,A-36:GR.19:F.I=0TO319:C.I:PL.20,0:DR.I,23:
N.I
60 POKE106,A-40:GR.21:F.I=1TO50:C.I:PL.39,24:DR.RND(0
)*79,RND(0)*47:N.I
140 GOS.DELAY
150 POKE106,A-36:GR.19+32
160 GOS.DELAY
170 POKE106,A-40:GR.21+32
```

To add a GRAPHICS 23 screen to this, add the following lines:

```
70 POKE106,A-48:GR.23:DEG:SE.0,15,12:F.I=1TO90
STEP3:C.I:PL.80,48:DR.SIN(I)*85,COS(I)*40
75 PL.80,48:DR.159-SIN(I)*85,95-COS(I)*40:N.I
180 GOS.DELAY
190 POKE106,A-48:GR.23+32
```

Now run the program and note how the pages are flipped. You can change the flipping rate by altering line 300.

Finally I have a 7-frame mini movie for you. Type in the program 'ACE MOVIE DIRECTOR', RUN it and sit back and watch for a few minutes.

The program is well structured so you shouldn't have any trouble understanding how it works if you have followed me so far.

Next time I'll have a complete utility which will allow you to draw or compose your own screens in different GRAPHICS or text modes. These will then be flipped at a speed of your choice to produce a mini movie. Till then....

Roberto Romano

```
1 REM ################################
2 REM #     ACE MOVIE DIRECTOR      #
3 REM #        by ROBERTO ROMANO    #
4 REM # Published by Atari Computer #
5 REM #     Enthusiasts (N.S.W.)    #
6 REM #          June 1984          #
7 REM ################################
10 P=PEEK(106):DELAY=370:TIME=200:A=3:
S=40
20 REM **************************
30 FOR LOOP=0 TO 6:POKE 106,P-16*LOOP:
C=4+LOOP:GOSUB 90:NEXT LOOP
40 REM **************************
50 FOR DISPLAY=0 TO 6:POKE 106,P-DISPL
AY*16:GOSUB DELAY:GRAPHICS 55:GOSUB 38
0:NEXT DISPLAY
60 TIME=TIME-10:IF TIME<10 THEN TIME=1
0
70 GOTO 50
80 REM *************************
90 GRAPHICS 23:GOSUB 380:FOR I=0 TO 36
0 STEP A
100 T=I/57.3:COLOR 1
110 GOTO C*30
120 Y=S*COS(4*SIN(2*T))*SIN(T)
130 X=S*COS(4*SIN(2*T))*COS(T)
140 GOTO 320
150 Y=S*SIN(4*SIN(2*T))*SIN(T)
160 X=S*SIN(4*SIN(2*T))*COS(T)
170 GOTO 320
180 Y=S*COS(SIN(8*T))*SIN(T)
190 X=S*COS(SIN(8*T))*COS(T)
200 GOTO 320
210 Y=S*COS(SIN(16*T))*SIN(T)
220 X=S*COS(SIN(16*T))*COS(T)
230 GOTO 320
240 Y=S*SIN(SIN(32*T))*SIN(T)
250 X=S*SIN(SIN(32*T))*COS(T)
260 GOTO 320
270 Y=S*SIN(COS(16*T))*SIN(T)
280 X=S*SIN(COS(16*T))*COS(T)
290 GOTO 320
300 Y=S*COS(4*SIN(4*T))*SIN(T)
310 X=S*COS(4*SIN(4*T))*COS(T)
320 IF I=0 THEN PLOT 80+X,48-Y
330 DRAWTO 80+X,48-Y
340 NEXT I
350 RETURN
360 REM *************************
370 FOR I=1 TO TIME:NEXT I:RETURN
380 HUE=INT(RND(0)*16):POKE 708,(HUE>0
)*(16*HUE-8)+(HUE=0)*56
390 RETURN
```

# NOTICES

***** 
A decision has been made on the costs of back issues of Inside Info. If the issues are still in stock they will cost $2.50 each. If they are no longer in stock and need to be photocopied they will be $4.00 each. These prices do not include postage. Contact the Club secretary for details on availability.

***** 
Your editor is now on-line. If you would like to grace the pages of Inside Info with an article, program or anything else just ring at a reasonable hour.

*****

***** 
We only received one response to the offer of bulk purchasing the Cicada modems. This has forced us to drop the offer.

***** 
No body has expressed any interest in the idea of bulk purchasing any products from overseas.

***** 
No interest has been shown in the idea of running special computer courses by the club.

*****

# PROGRAM REVIEW

In Inside Info #11 (and again in this issue), I used several screen dumps to fill in some of the spaces. These screen dumps were 'dumped' to my Epson printer using a program called MEGAFONT ($US19.95) which was written by Randy Dellinger and Richard Rognlie.

The pictures themselves were from the MegaFont Picture Disk #1 ($US10.00) which were composed by Mark Whiteside and Jennifer Brabson. I should mention that these pictures are copyrighted and have been used by permission -my thanks to the authors.

## MEGAFONT

MegaFont is a compiled BASIC program which comes on an AUTOBOOT disk and can be used on suitable EPSON (Graftrax) or PROWRITER/NEC printers. It has 4 options:-

1.  Convert Atari resident character set.
2.  LIST files with all ATASCII characters.
3.  Dump GRAPHICS 7+ or 8 screen.
4.  End program.

### 1. Convert Character Set

This option allows you to read a character set file which was created using any Character Editor such as INSTEDIT, Graphics Master or SuperFont (from Compute!'s First Book of Atari Graphics).

The original file of 256 characters is read and converted. After about 90 seconds, a new file is created and written to disk, ready for you to use. So any font you can define for use on your Atari can be converted for use on your printer. The cover of this issue gives some examples.

### 2. Special LISTer

This option accesses any disk-file and allows you to print it using any font that you create through option 1. Two problems that I've had while using MegaFont are that:-

* it doesn't 'page' your printout, so you can't use seperate sheets of paper.
* it prints straight across the page without doing any formatting, therefore any word-processing files will be printed showing all text formatting blocks as well as text.

There are however, a few word-processors which will "print to disk", that is, it will insert all necessary blanks and format the file as it will be printed on paper -Text Wizard and the old Atari Word-Processor have this capability.

To print BASIC program listings the program must be LISTed to disk, otherwise MegaFont will print-out the 'tokenized' BASIC form of the program. Once again, the program doesn't format the listing for you.

To do this I have written my own formatting program which reads a LISTed file, inserts appropriate blanks, pages it (as seperate disk-files) and then writes it back to disk. I have used this option to print-out the MILLIONARE listing in this issue because that program contains several special characters.

There are two font sizes available through this option, both of which are larger than the usual font sizes on the printer, hence any text files or program listings take up more space than the usual condensed font.

NOTES: MegaFont does not down-load the character set to the printer, but rather, converts the characters in memory and 'dumps' them to the printer.

If you've tried to redefine the character set on the Epson FX-80, you'll know that the single-byte 'control codes' can't be user defined.

### 3. Graphics Dumper

This option allows you to dump a GRAPHICS 7+ or GRAPHICS 8 screen to your printer. A program listing is provided which will read your screen, creating a disk-file of 62 sectors, which can then be read through this option for dumping to the printer.

You can print your screen in either, INVERSE or NORMAL mode, as a quarter page, half page or full page -the full page size being printed side ways on the printer. For those who use MicroPainter this program is ideal for printing the pictures you create.

### MegaFont II

I have just received word that MegaFont II has been released. The program now comes in a binder for $US24.95. Version II has had its documentation updated and expanded and includes two new ready to run fonts. Also, it now allows you to:-
* type directly to printer using any of your special fonts.
* select the number of line feeds and use single sheet paper.

MegaFont II now includes instructions on how it can be used in conjunction with the Koalapad and AtariWriter as well as the new XL computers.

### MegaFont Picture Disk 1

This disk contains 11 different pictures, many of which were created using MicroPainter.
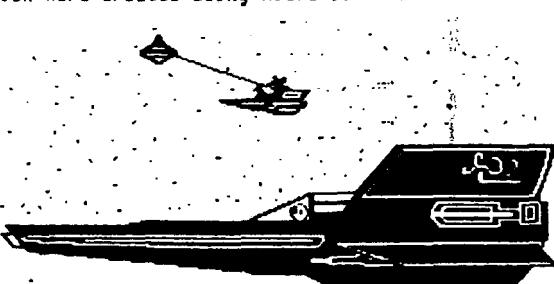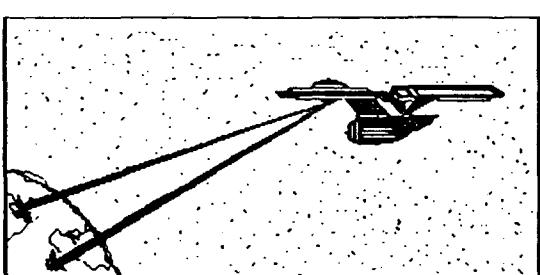
# THE MILLIONAIRE

by Jamie Athas. (Maroubra, N.S.W.)

This is another LOTTO NUMBER SELECTOR PROGRAM, but with a difference -it talks! The MILLIONAIRE makes use of SAM, the Software Automatic Mouth and it could make you a millionaire.

## INSTRUCTIONS

1 Type the program EXACTLY as shown then SAVE the program to cassette or disk.

2 Turn the computer off then boot the 'SAM' program. If you don't have SAM then don't worry -the program still runs but without any voice.

3 LOAD "The Millionaire" program and run it. Then SAM will start to speak his first words, "I'm picking my lotto numbers."

The computer will then draw up one game of a LOTTO coupon and fill it out, showing the selected numbers. If SAM is present the numbers will be spoken.

You can then press either:-

SPACE BAR for the computer repeat, through speech, which numbers were selected, OR

RETURN to select a new set of numbers.

I hope to write a lot more software using SAM in future issues of INSIDE INFO.

In the event that you do win any money using the MILLIONAIRE, I'll expect a 10% cut of the prize for my troubles. HA,HA,HA.

## Keystroke Instructions

The following table shows all the necessary keystrokes which are required to get the special ATASCII characters which have been used in Jamie's "MILLIONARE" program.

| ATASCII Number | ATASCII Character | Necessary Keystrokes |
|---|---|---|
| CHR$(17) | ⌐ | CTRL-Q |
| CHR$(18) | — | CTRL-R |
| CHR$(5) | ⌐ | CTRL-E |
| CHR$(124) | ¦ | CTRL-V |
| CHR$(26) | ∟ | CTRL-Z |
| CHR$(3) | ⌐ | CTRL-C |
| CHR$(125) | ⌐ | ESC then SHIFT-CLEAR |
| CHR$(160) | ■ | INVERSE SPACE |

```
1 REM ###########################################
2 REM #        THE MILLIONAIRE           #
3 REM #        by Jamie Athas            #
4 REM # Published by Atari Computer      #
5 REM #      Enthusiasts (N.S.W.)        #
6 REM #          June 1984               #
7 REM ###########################################
10 SAM=8192:DIM SAM$(150):GOTO 410
20 A=INT(RND(0)*40)+1
30 B=INT(RND(0)*40)+1
40 IF A=B THEN 30
50 C=INT(RND(0)*40)+1
60 IF C=B OR C=A THEN 50
70 D=INT(RND(0)*40)+1
80 IF D=C OR D=B OR D=A THEN 70
90 E=INT(RND(0)*40)+1
100 IF E=D OR E=C OR E=B OR E=A THEN 90
110 F=INT(RND(0)*40)+1
120 IF F=E OR F=D OR F=C OR F=B OR F=A THEN 110
130 POKE 82,0:DL=PEEK(560)+PEEK(561)*256:POKE DL+3,70:POKE DL+6,6
140 ? "K":POKE 710,12:POKE 709,8:POKE 712,12:POSITION 0,0:? " this weeks LOTTO NUMBERS ARE"
150 POSITION 0,2:? ,"⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐"
160 ? ,"|1 ||2 ||3 ||4 ||5 ||6 |"
170 ? ,"∟∟ ∟∟ ∟∟ ∟∟ ∟∟ ∟∟"
180 ? ,"⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐"
190 ? ,"|7 ||8 ||9 ||10||11||12|"
200 ? ,"∟∟ ∟∟ ∟∟ ∟∟ ∟∟ ∟∟"
210 ? ,"⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐"
220 ? ,"|13||14||15||16||17||18|"
230 ? ,"∟∟ ∟∟ ∟∟ ∟∟ ∟∟ ∟∟"
240 ? ,"⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐"
250 ? ,"|19||20||21||22||23||24|"
260 ? ,"∟∟ ∟∟ ∟∟ ∟∟ ∟∟ ∟∟"
270 ? ,"⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐"
280 ? ,"|25||26||27||28||29||30|"
290 ? ,"∟∟ ∟∟ ∟∟ ∟∟ ∟∟ ∟∟"
300 ? ,"⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐"
310 ? ,"|31||32||33||34||35||36|"
320 ? ,"∟∟ ∟∟ ∟∟ ∟∟ ∟∟ ∟∟"
330 ? ,"⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐ ⌐⌐⌐"
340 ? ,"|37||38||39||40|"
350 ? ,"∟∟ ∟∟ ∟∟ ∟∟"
360 SAM$=",":GOSUB 1000+A:GOSUB 1000+B:GOSUB 1000+C:GOSUB 1000+D:GOSUB 1000+E:GOSUB 1000+F:GOTO 450
370 POKE 764,255
380 IF PEEK(764)=12 THEN 410
390 IF PEEK(764)=33 THEN 450
400 GOTO 380
410 GRAPHICS 0:POKE 710,0:POKE 712,53:POKE 752,1
420 POSITION 3,10:? "I'm picking Lotto numbers!"
430 FOR OP=1 TO 200:NEXT OP:SAM$=",",AY8
```

```
M PIH4KIHNG MAY  LAA4TOH WAH4MBERS,":X
=USR(SAM)
440 GOTO 20
450 X=USR(SAM):GOTO 370
1001 POSITION 11,3:? "[1]":SAM$(LEN(SAM
$)+1)="WAH4N,":RETURN
1002 POSITION 15,3:? "[2]":SAM$(LEN(SAM
$)+1)="TUW4,":RETURN
1003 POSITION 19,3:? "[3]":SAM$(LEN(SAM
$)+1)="THRIY4,":RETURN
1004 POSITION 23,3:? "[4]":SAM$(LEN(SAM
$)+1)="FOH4R,":RETURN
1005 POSITION 27,3:? "[5]":SAM$(LEN(SAM
$)+1)="FAY4V,":RETURN
1006 POSITION 31,3:? "[6]":SAM$(LEN(SAM
$)+1)="SIH4KS,":RETURN
1007 POSITION 11,6:? "[7]":SAM$(LEN(SAM
$)+1)="SEH4VIXN,":RETURN
1008 POSITION 15,6:? "[8]":SAM$(LEN(SAM
$)+1)="EY4T,":RETURN
1009 POSITION 19,6:? "[9]":SAM$(LEN(SAM
$)+1)="NAY4N,":RETURN
1010 POSITION 23,6:? "[10]":SAM$(LEN(SAM
$)+1)="TEH4N,":RETURN
1011 POSITION 27,6:? "[11]":SAM$(LEN(SAM
$)+1)="IXLEH4VIXN,":RETURN
1012 POSITION 31,6:? "[12]":SAM$(LEN(SAM
$)+1)="TWEH4LV,":RETURN
1013 POSITION 11,9:? "[13]":SAM$(LEN(SAM
$)+1)="THER4TIY6N,":RETURN
1014 POSITION 15,9:? "[14]":SAM$(LEN(SAM
$)+1)="FOH4TIY6N,":RETURN
1015 POSITION 19,9:? "[15]":SAM$(LEN(SAM
$)+1)="FIHFTIY6N,":RETURN
1016 POSITION 23,9:? "[16]":SAM$(LEN(SAM
$)+1)="SIH4KSTIY6N,":RETURN
1017 POSITION 27,9:? "[17]":SAM$(LEN(SAM
$)+1)="SEH4VIXNTIY6N,":RETURN
1018 POSITION 31,9:? "[18]":SAM$(LEN(SAM
$)+1)="EY4TIY6N,":RETURN
1019 POSITION 11,12:? "[19]":SAM$(LEN(SA
M$)+1)="NAY4NTIY6N,":RETURN
1020 POSITION 15,12:? "[20]":SAM$(LEN(SA
M$)+1)="TWEH4NTIY,":RETURN
1021 POSITION 19,12:? "[21]":SAM$(LEN(SA
M$)+1)="TWEH4NTIY WAH4N,":RETURN
1022 POSITION 23,12:? "[22]":SAM$(LEN(SA
M$)+1)="TWEH4NTIY TUW4,":RETURN
1023 POSITION 27,12:? "[23]":SAM$(LEN(SA
M$)+1)="TWEH4NTIY THRIY4,":RETURN
1024 POSITION 31,12:? "[24]":SAM$(LEN(SA
M$)+1)="TWEH4NTIY FOH4R,":RETURN
1025 POSITION 11,15:? "[25]":SAM$(LEN(SA
M$)+1)="TWEH4NTIY FAY4V":RETURN
1026 POSITION 15,15:? "[26]":SAM$(LEN(SA
M$)+1)="TWEH4NTIY SIH4KS,":RETURN
1027 POSITION 19,15:? "[27]":SAM$(LEN(SA
M$)+1)="TWEH4NTIY SEH4VIXN,":RETURN
1028 POSITION 23,15:? "[28]":SAM$(LEN(SA
M$)+1)="TWEH4NTIY EY4T,":RETURN
1029 POSITION 27,15:? "[29]":SAM$(LEN(SA
M$)+1)="TWEH4NTIY NAY4N,":RETURN
1030 POSITION 31,15:? "[30]":SAM$(LEN(SA
M$)+1)="THER4TIY,":RETURN
1031 POSITION 11,18:? "[31]":SAM$(LEN(SA
M$)+1)="THER4TIY WAH4N,":RETURN
1032 POSITION 15,18:? "[32]":SAM$(LEN(SA
M$)+1)="THER4TIY TUW4,":RETURN
1033 POSITION 19,18:? "[33]":SAM$(LEN(SA
M$)+1)="THER4TIY THRIY4,":RETURN
1034 POSITION 23,18:? "[34]":SAM$(LEN(SA
M$)+1)="THER4TIY FOH4R,":RETURN
1035 POSITION 27,18:? "[35]":SAM$(LEN(SA
M$)+1)="THER4TIY FAY4V,":RETURN
1036 POSITION 31,18:? "[36]":SAM$(LEN(SA
M$)+1)="THER4TIY SIH4KS,":RETURN
1037 POSITION 11,21:? "[37]":SAM$(LEN(SA
M$)+1)="THER4TIY SEH4VIXN,":RETURN
1038 POSITION 15,21:? "[38]":SAM$(LEN(SA
M$)+1)="THER4TIY EY4T,":RETURN
1039 POSITION 19,21:? "[39]":SAM$(LEN(SA
M$)+1)="THER4TIY NAY4N,":RETURN
1040 POSITION 23,21:? "[40]":SAM$(LEN(SA
M$)+1)="FOH4RTIY,":RETURN
```

# NOTICES

*****

It's time for you to cast your vote for the article you feel was the best out of all the articles printed in Inside Info issues 7 to 12 (inclusive). The winner will receive $100 prizemoney to spend on whatever they like. Voting will take place at the Club meeting to be held on 2nd of July. If you can't get there mail your vote in time to be counted!!!
PLEASE NOTE: The article needn't necessarily contain a program listing.

*****

At the present time our club is about $3,700.00 in the black. The Committee feels that it would be wise to earn some interest on about $2000 of this money by putting it into an interest bearing account. This money may be needed for unforeseen expenses -e.g. if an unthinking member were to steal something from our OTC meeting place we'd get kicked out. We would then need to find a meeting place we'd have to pay for, and that could be very expensive! Hopefully it'll never happen. The rest of the money would remain in the present cheque account to cover operating expenses and club purchases. Let us know your feelings on this.

*****

The introduction of DOS 3 has caused a few problems. Some of my utilities for programming are DOS 2 and can't be converted for DOS 3. Until further notice, anything you send to the club on disk should be in DOS 2 format. While DOS 2 files can be converted to DOS 3, it is awkward to convert the other way.

While on the subject of DOS 3, I mentioned last issue that Futuretronics would send you a copy of DOS 3 if you asked them for it. They later advised that you need to send them a blank disk first. I should also mention that you won't receive either of the two manuals which normally come with DOS 3! (real handy for beginners).

*****

# DISKussion

## by Ken Scalley. (Winston Hills, N.S.W.)

This month I am going to digress a bit and talk about a menu program instead of disk layout,etc.

A few weeks ago, a member of our user group who lives near me, Bob Penney, brought a menu program around to show me -he had punched it in from the September, 1983 issue of Antic and we both agreed that it could do with a few enhancements.

It was the usual kind of menu program..you put it on your disks as an AUTORUN.SYS file so that when you boot your disk, the menu program displays all (or some) of the binary files on the disk. You can then load a program with just one keystroke.

With any AUTORUN.SYS file on your disk, you don't need to have the DUP.SYS file present -it is just a waste of precious disk space. This was mentioned the other month by our editor (see Odds and Sods, page 19 -Inside Info #11). He said that the only way to get DOS.SYS on a disk by itself, without DUP.SYS, was to use option "H" (write DOS files) in the DOS menu and to then delete DUP.SYS, using option "D". I thought this was right, and that it was silly that you couldn't write DOS.SYS on its own -I figured that there had to be an easier way. Well, proving the law of Atari computers (you know, there is more not discovered than is already known) I read in my Eugene ACE newsletter that the following will do what is required:-

Type in, in immediate mode (no line numbers),

        END:OPEN #1,8,0,"D:DOS.SYS":END

and this will write DOS.SYS onto any formatted disk! Getting back to the menu program....as it stood, it suffered from several problems:-

1) You couldn't re-run the program without re-booting the disk.

2) It wouldn't load some binary load files and would cause the infamous Atari "system lock-up".

3) We didn't like the colours used (a personal matter,admittedly).

So, having started to play around with assembler recently (and in the process of growing an extra 6 fingers to learn hex (maybe that's what our two-fingered editor needs??) I decided to change the program and overcome the above inadequacies. The resulting program has these features:-

1) Pressing START or SYSTEM RESET will re-run the program without re-booting.

2) It will load all of the binary files that I could find to try.

3) It now uses the colours that I like.

### LISTING 1

RUNning the program given in listing 1 (below) will write the appropriate AUTORUN.SYS file to your disk. Please note; this is a "no frills" program...if you already have an AUTORUN.SYS file on the disk (and it is not locked), it will be overwritten without any warning! -If the file is locked, the program will crash.

Having written this program, I then had to put it onto all of my disks of binary load files. Using DOS, this entailed checking if there was a locked file already there, unlocking it, loading the file into memory, and writing to disk (for each disk)...so I decided to write a little program to do this more easily.

### LISTING 2

Listing 2 will read any file once, and allow you to write it to as many disks as you want; provided the file will fit into the programs buffer. On each occasion, it will give you the opportunity to write with or without verify, as well as the choice of whether or not you want to lock the file after writing it. It also checks for an existing file on the destination disk and asks you if you want to overwrite it. If you want to write DOS.SYS the file does not need to load, rather, it will be written in the manner described above. It is not possible to overwrite an existing DOS.SYS file, though you can lock it with this program.

Well, that's it for another month. While I am enjoying writing this column, I find it hard without any feedback from readers....does anyone read this column???

......do you love it??

......do you hate it???

...would you like a discussion on some specific disk problem????

Please!!!!!!!!!!!some feedback!!!!!

[Editor's Note: You can write to Ken through the editor C/- the club address. Please show him your support and interest -as I've already stated, we can't cater for your needs if we don't know what you want!]

---

## LISTING 1

```
1 REM ##################################
2 REM #         AUTOMENU              #
3 REM #       by Ken Scalley          #
4 REM # Published by Atari Computer   #
5 REM #    Enthusiasts (N.S.W.)       #
6 REM #          June 1984            #
7 REM ##################################
10 ? "INSERT DISK,PUSH RETURN":POKE 76
4,255
20 IF PEEK(764)=255 THEN 20
30 POKE 764,255
40 OPEN #1,8,0,"D:AUTORUN.SYS"
50 PUT #1,255:PUT #1,255:PUT #1,0:PUT
#1,56:PUT #1,144:PUT #1,59
60 READ B:IF B<>-1 THEN PUT #1,B:GOTO
60
70 PUT #1,224:PUT #1,2:PUT #1,225:PUT
#1,2:PUT #1,0:PUT #1,56
80 CLOSE #1
90 DATA 173,255,54,201,0,240
100 DATA 3,32,64,21,169,1
110 DATA 141,255,54,169,0,133
120 DATA 12,169,56,133,13,169
130 DATA 1,133,9,169,0,141
140 DATA 68,2,169,0,133,203
150 DATA 169,55,133,204,133,206
160 DATA 32,163,58,169,0,141
170 DATA 198,2,238,240,2,160
180 DATA 0,132,86,132,207,152
190 DATA 200,132,84,200,145,88
```

```
200 DATA 169,6,133,85,133,82
210 DATA 169,31,133,83,162,96
220 DATA 169,9,157,66,3,169
230 DATA 204,157,68,3,169,58
240 DATA 157,69,3,169,79,157
250 DATA 72,3,32,86,228,169
260 DATA 2,133,82,169,39,133
270 DATA 83,169,20,133,84,169
280 DATA 8,133,85,162,96,169
290 DATA 27,157,68,3,169,59
300 DATA 157,69,3,169,23,157
310 DATA 72,3,32,86,228,169
320 DATA 21,133,84,169,9,133
330 DATA 85,169,50,157,68,3
340 DATA 169,59,157,69,3,169
350 DATA 20,157,72,3,32,86
360 DATA 228,162,16,169,3,157
370 DATA 66,3,169,123,157,68
380 DATA 3,169,59,157,69,3
390 DATA 169,6,157,74,3,169
400 DATA 0,157,75,3,32,86
410 DATA 228,162,16,169,7,157
420 DATA 66,3,169,238,157,68
430 DATA 3,169,55,157,69,3
440 DATA 169,18,157,72,3,169
450 DATA 0,157,73,3,32,86
460 DATA 228,48,46,160,10,162
470 DATA 1,185,238,55,221,138
480 DATA 59,208,212,232,200,192
490 DATA 13,208,242,160,0,162
500 DATA 2,189,238,55,145,203
510 DATA 232,200,192,8,208,245
520 DATA 230,207,165,207,201,26
530 DATA 240,5,32,194,58,208
540 DATA 178,162,16,169,12,157
550 DATA 66,3,32,86,228,165
560 DATA 207,208,3,76,120,58
570 DATA 169,6,133,84,169,1
580 DATA 133,225,169,0,133,203
590 DATA 169,32,160,0,153,208
600 DATA 55,200,192,30,208,248
610 DATA 169,5,133,85,165,207
620 DATA 41,1,133,226,165,207
630 DATA 74,24,101,226,133,224
640 DATA 165,225,24,105,192,141
650 DATA 208,55,169,174,141,209
660 DATA 55,160,0,162,3,177
670 DATA 203,157,208,55,200,232

680 DATA 192,8,208,245,165,226
690 DATA 240,6,166,224,228,225
700 DATA 240,43,173,208,55,24
710 DATA 101,224,141,227,55,169
720 DATA 174,141,228,55,160,0
730 DATA 162,22,165,204,133,206
740 DATA 165,224,10,10,10,24
750 DATA 101,203,133,205,177,205
760 DATA 157,208,55,200,232,192
770 DATA 8,208,245,162,96,169
780 DATA 9,157,66,3,169,208
790 DATA 157,68,3,169,55,157
800 DATA 69,3,169,30,157,72
810 DATA 3,32,86,228,32,194
820 DATA 58,230,225,165,225,197
830 DATA 224,176,3,76,44,57
840 DATA 208,3,76,44,57,169
850 DATA 255,141,252,2,173,252
860 DATA 2,201,255,208,10,173
870 DATA 31,208,201,6,208,242
880 DATA 76,15,56,160,0,185
890 DATA 97,59,205,252,2,240
900 DATA 33,200,196,207,208,243
910 DATA 169,253,32,164,246,169
920 DATA 245,141,198,2,169,0
930 DATA 133,20,165,20,201,21
940 DATA 208,250,169,0,141,198
950 DATA 2,76,199,57,162,0
960 DATA 142,198,2,152,152,10
970 DATA 10,10,133,203,160,0
980 DATA 162,2,177,203,201,32
990 DATA 240,9,157,128,59,200
1000 DATA 232,192,8,208,241,160
1010 DATA 0,185,138,59,157,128
1020 DATA 59,200,232,192,14,208
1030 DATA 244,162,96,169,12,157
1040 DATA 66,3,32,86,228,32
1050 DATA 163,58,162,16,169,3
1060 DATA 157,66,3,169,128,157
1070 DATA 68,3,169,59,157,69
1080 DATA 3,169,0,157,73,3
1090 DATA 169,4,157,74,3,32
1100 DATA 86,228,169,255,141,252
1110 DATA 2,169,47,141,224,2
1120 DATA 169,16,141,225,2,169
1130 DATA 0,133,84,133,85,76
1140 DATA 200,21,169,12,133,84
1150 DATA 169,7,133,85,162,96

1160 DATA 169,9,157,66,3,169
1170 DATA 70,157,68,3,169,59
1180 DATA 157,69,3,169,27,157
1190 DATA 72,3,32,86,228,173
1200 DATA 31,208,201,6,208,249
1210 DATA 76,15,56,162,96,169
1220 DATA 3,157,66,3,169,202
1230 DATA 157,68,3,169,58,157
1240 DATA 69,3,169,12,157,74
1250 DATA 3,169,0,157,75,3
1260 DATA 32,86,228,96,24,165
1270 DATA 203,105,8,133,203,96
1280 DATA 83,58,125,14,14,14
1290 DATA 14,14,14,14,14,14
1300 DATA 14,14,14,14,14,14
1310 DATA 14,14,14,14,14,14
1320 DATA 14,14,14,14,14,22
1330 DATA 66,73,78,65,82,89
1340 DATA 32,76,79,65,68,32
1350 DATA 77,69,78,85,32,80
1360 DATA 82,79,71,82,65,77
1370 DATA 2,13,13,13,13,13
1380 DATA 13,13,13,13,13,13
1390 DATA 13,13,13,13,13,13
1400 DATA 13,13,13,13,13,13
1410 DATA 13,13,13,112,117,115
1420 DATA 104,32,107,101,121,32
1430 DATA 116,111,32,108,111,97
1440 DATA 100,32,97,32,103,97
1450 DATA 109,101,112,117,115,104
1460 DATA 32,83,84,65,82,84
1470 DATA 32,116,111,32,114,101
1480 DATA 45,114,117,110,78,79
1490 DATA 32,34,79,66,74,34
1500 DATA 32,70,73,76,69,83
1510 DATA 32,79,78,32,84,72
1520 DATA 73,83,32,68,73,83
1530 DATA 75,63,21,18,58,42
1540 DATA 56,61,57,13,1,5
1550 DATA 0,37,35,8,10,47
1560 DATA 40,62,45,11,16,46
1570 DATA 22,43,23,68,58,42
1580 DATA 46,42,68,58,32,32
1590 DATA 32,32,32,32,32,32
1600 DATA 46,79,66,74,0,0
1610 DATA 0
1620 DATA -1
```

## LISTING 2

```
1 REM #############################
2 REM #          MULTICOPY        #
3 REM #       by Ken Scalley       #
4 REM # Published by Atari Computer #
5 REM #   Enthusiasts (N.S.W.)     #
6 REM #         June 1984          #
7 REM #############################
10 GRAPHICS 0:POKE PEEK(560)+256*PEEK(
561)+9,64:POKE PEEK(560)+256*PEEK(561)
+23,64:POKE 710,0:POKE 712,80
20 ? "    DOS MULTI-COPY UTILITY":?
"     version 1   DOS 2.0":? :? "
   Written by Ken Scalley"
30 REM SET UP
40 POKE 752,1:POSITION 14,5:? "verify
on":POSITION 12,14:? "setting up...":D
IM FILEIN$(14),LOCK$(3),B$(38)
50 B$=" ":B$(38)=" ":B$(2)=B$:SPACE=FR
E(0)-2000:DIM BUFFER$(SPACE):DFF=40000
:LOCK$="OFF"
60 POKE 1913,87:POSITION 8,7:? "lock a
fter writing ";LOCK$
70 RESTORE :FOR J=1700 TO 1751:READ X:
POKE J,X:NEXT J
80 DATA 32,68,218,104,201,4,208,43,104
,104,10,10,10,10,170,104,104,157,66,3,
104,157,69,3,104,157,68,3,104,157
90 DATA 73,3,104,157,72,3,32,86,228,13
```

```
2,212,189,72,3,133,203,189,73,3,133,20
4,96
100 DATA 169,255,133,240,169,255,133,2
41,162,0,160,0,177,240,133,255,160,2,1
77,240,133,253,133,251,200,177
110 DATA 240,133,254,133,252,200,177,2
40,133,249,200,177,240,133,250,169,6,1
01,240,144,2,230,241,133,240,160
120 DATA 0,177,240,145,253,200,208,249
,230,254,230,241,198,255,208,241,108,2
51,0,234
130 REM FILENAME TO LOAD FROM
140 POSITION 2,14:? B$:POSITION 2,10:?
B$:GOSUB 610:POSITION 2,18:? "insert
```

16

```
source disk"                          -toggle lock":POKE 764,255              writing data to file...
150 ? "name of file to copy";CHR$(253)  310 REM VERIFY? & LOCK?               510 TRAP 630:CLOSE #1:OPEN #1,8,0,FILE
:TRAP 140:POSITION 22,19:INPUT FILEIN$  320 IF PEEK(764)=12 THEN 420          IN$:TRAP OFF
:TRAP OFF                              330 IF PEEK(53279)=3 THEN 370         520 REM WRITE TO FILE
160 IF FILEIN$="" THEN 150             340 IF PEEK(53279)=5 THEN 390         530 IF FLAG THEN CLOSE #1:GOTO 580
170 FOR J=1 TO LEN(FILEIN$):IF FILEIN$ 350 IF PEEK(53279)=6 THEN RUN          540 POSITION 2,14:? "    writing dat
(J,J)="?" OR FILEIN$(J,J)="*" THEN POP 360 GOTO 320                           a to file...   ":X=USR(1700,1,11,ADR(B
:GOTO 140                             370 IF PEEK(1913)=80 THEN POKE 1913,87  UFFER$),SPACE)
180 IF LEN(FILEIN$)<3 THEN 200         :POSITION 21,5:? "on ":FOR J=1 TO 30:N 550 CLOSE #1:POSITION 2,14:? B$:GOSUB
190 IF FILEIN$(1,2)="D:" THEN 210      EXT J:GOTO 320                         610
200 FOR J=LEN(FILEIN$) TO 1 STEP -1:FI 380 IF PEEK(1913)=87 THEN POKE 1913,80  560 IF X<>1 THEN POSITION 2,14:? "
LEIN$(J+2,J+2)=FILEIN$(J,J):NEXT J:FIL :POSITION 21,5:? "off":FOR J=1 TO 30:N error writing to file(";X;")";CHR$(2
EIN$(1,2)="D:"                         EXT J:GOTO 320                         53):GOTO 290
210 POSITION 7,10:? "filename- ";FILEI 390 IF LOCK$="OFF" THEN LOCK$="ON ":PO 570 REM LOCK FILE
N$;B$:IF FILEIN$="D:DOS.SYS" THEN FLAG SITION 27,7:? LOCK$:FOR J=1 TO 30:NEXT 580 IF LOCK$="ON " THEN POSITION 2,14:
=1:GOTO 290                            J:GOTO 320                             ? "       locking file...";B$:XIO 35
220 TRAP 140:CLOSE #1:OPEN #1,4,0,FILE 400 IF LOCK$="ON " THEN LOCK$="OFF":PO ,#1,0,0,FILEIN$
IN$:TRAP OFF                           SITION 27,7:? LOCK$:FOR J=1 TO 30:NEXT 590 REM WRITTEN!
230 REM LOAD FILE INTO BUFFER          J:GOTO 320                             600 POSITION 2,14:? "        file wri
240 POSITION 2,14:? "    loading dat   410 REM CHECK DISKFILE                 tten";B$:GOTO 280
a from file...":GOSUB 610:X=USR(1700,1 420 TRAP 630:CLOSE #1:OPEN #1,4,0,FILE 610 POSITION 2,17:? B$;B$;B$;B$;B$:POS
,7,ADR(BUFFER$),SPACE)                 IN$:CLOSE #1:POSITION 2,14:? "     fi  ITION 2,17:RETURN
250 POSITION 2,14:? B$:IF X=1 THEN POS le already exists!      "              620 REM ERROR OPENING FILE
ITION 2,14:? " insufficient memory t   430 IF FLAG THEN 580                   630 CLOSE #1:GOSUB 610:IF PEEK(195)=17
o load file";CHR$(253):END             440 GOSUB 610:? :? "        REPLACE FIL 0 THEN 500
260 IF X<>136 THEN POSITION 2,14:? "   E? (Y/N)":POKE 764,255                640 IF PEEK(195)=167 THEN 670
error loading data from file(";X;")";  450 IF PEEK(764)=35 THEN POSITION 2,14 650 POSITION 2,14:? "      error open
CHR$(253):END                          :? B$:GOTO 290                         ing file(";PEEK(195);")";CHR$(253):GOT
270 SPACE=PEEK(203)+256*PEEK(204)      460 IF PEEK(764)=43 THEN 490           0 290
280 REM GET DESTINATION DISK IN DRIVE  470 GOTO 450                           660 REM UNLOCK FILE
290 GOSUB 610:? "insert destination di 480 REM OPEN FILE                      670 POSITION 2,14:? "      unlocking
sk, press RETURN";CHR$(253):? :? "STAR 490 POSITION 2,14:? "     deleting o   old file...    ":TRAP 490:XIO 36,#1,0
T -RUN"                                ld file...";B$                        ,0,FILEIN$:GOTO 490
300 ? "OPTION-toggle verify":? "SELECT 500 IF FLAG THEN POSITION 2,14:? "
```



# BOOK REVIEW
## 6502 ASSEMBLY LANGUAGE PROGRAMMING
### BY LANCE A. LEVENTHAL
(PUBLISHED BY OSBORNE/MCGRAW HILL)
Reviewed by Ken Shiu. (Lugarno, N.S.W.)

©1983 JBB

After being very disappointed with "Programming the 6502" (refer review in Inside Info #12 page 16), I turned to Lance Leventhal's, 6502 Assembly Language Programming which is one of the books recommended in the Atari Users Manual. This book is excellent, 600 pages chock full of information, tables and diagrams -suitable for anyone from beginner to expert. An advantage to the beginner would be that the book is easy to understand, although the beginner will have some difficulty understanding the text devoted to the advanced readers.

The beginning of the book introduces the reader to the world of the 6502 -assemblers, the instruction set, simple programs, character codes (e.g. ASCII, Hex.), arithmetic operands, etc.

In catering for the intermediate and advanced reader it gives a detailed study into 6502, Input/Output and interfacing, interupts, program design and debugging techniques. Leventhal has also included two sample hardware projects -a Digital stopwatch and thermometer.

An excellent feature of this book is that the important points and concepts are printed in boldface with the lighter type elaborating on the boldface.

Every example is set out in a clear and concise manner. An aim, sample problems, flowchart, source program, subroutine documentation, object program and a very detailed line by line explanation are given to complement every example. Very Neat.

In general, I found this to be an excellent book. A definite must for any assembler programmer. I highly recommend it!
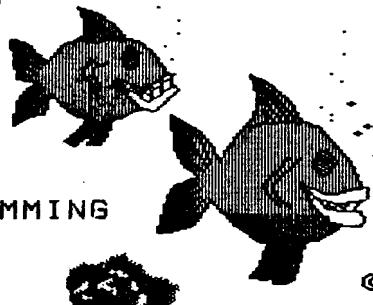
# SOFTWARE EXCHANGE PROGRAMS

from Brian Simmons (Software Exchange)

Many people have been requesting a list of the software which is available through the Software Exchange -here it is! Disks cost $10 each and Tapes cost $8 each with instructions supplied.

PLEASE NOTE:- Not all of these programs are available on tape.

## ACE NSW Disk/Tape #1
### UTILITIES #1

**MASSACRE** Allows you to selectively delete multiple files from a diskette, even if they're locked. Ideal for 'cleaning up' those old disks.

**MEMTEST** A machine language program used to test your ATARI's Random Access Memory.

**NUM.OBJ** Provides the BASIC user with automatic line number generation.

**MICROASM.B01** This is an assembler which can:-
1) Input and assemble source code.
2) Save object code on tape.
3) Load object code from tape.
4) Execute the object program (USR).
5) Call the object program and pass a parameter.
6) Disassemble an object program to the screen.

**ASM (SYMBOLIC ASSEMBLER)** This assembler is written in BASIC, is disk oriented and supports symbols and labels.

**PRINTNOP** This is the assembler source code for a parallel printer driver which utilizes the ATARI's joystick ports (Nos. 3 & 4).

## ACE NSW Disk/Tape #2
### GAMES #1

**ARTILLER.BAS** A game for 2 people. The object is to blow up your opponent by firing an artillery shell while judging the angle of your gun and the amount of gun powder required.

**BLACKJACK** Card game played against the computer.

**BOING** The object of this game is to bounce on a trampoline for the greatest number of times without falling off.

**APPLE CIDER** You must protect your apples from being squashed by the invading alien ship.

**DOGGIES** A game of logic. You have to arrange in correct order, six little doggies in the least number of moves.

**GOBLIN** Like PACMAN (tm) but there's only one ghost.

**CLEWSO** You assume the role of the famous inspector Closeau to solve a murder case involving five suspects.

**HANGMAN** The traditional spelling game of 'guess the word'.

**CHICKEN** Make the chicken cross the busy Highway without getting run over!

**LINES** An addictive two player game in which you to try to get your opponent to run into your line, the border or himself.

## ACE NSW Disk #3
## "ATARI figFORTH 1.4S"

The language "FORTH", available only on disk.

## ACE NSW Disk/Tape #4
## "INSIDE INFO VOLUME ONE"

Ready to run programs that featured in issues 1 to 6 of Inside Info.

## ACE NSW Disk/Tape #5
## "INSIDE INFO VOLUME TWO"

Ready to run programs as featured in issues 7 to 10 of Inside Info. This disk also contains the cartoons (by Allan R. Smyth) which were printed in Inside Info #12 (April 1984).

## ACE NSW Disk #6
## XL TRANSLATOR DISK

Several popular programs won't run on the new XL machine. This disk has two operating systems (one per side) which will allow you to run virtually any existing software.

---

# 'COMPUTE!' PROGRAM EXCHANGE
###### ****** (Refer to page 7) ******
## LIBRARY CONTENTS:

| NAME | DESCRIPTION | ISSUE | NAME | DESCRIPTION | ISSUE | NAME | DESCRIPTION | ISSUE |
|---|---|---|---|---|---|---|---|---|
| MEMDUMP | SELF EXPLAINS | MAR '81 | TAG | GAME | OCT '82 | WEDGE | SCREEN PRINTER | MAY '83 |
| INVADERS | GAME | APR '81 | METSTORM | GAME | OCT '82 | RESETRUN | RELOCATE PRGM. | JUN '83 |
| BLOCKADE | GAME | AUG '81 | SUPCHASE | GAME | OCT '82 | SORT USR | UTILITY ROUT. | OCT '83 |
| PONG | GAME | SEP '81 | LASERGUN | GAME | NOV '82 | ANTAROBI | GAME | OCT '83 |
| ISLNDJMP | GAME | DEC '81 | UXB | GAME | NOV '82 | DETECT. | STR$ SEARCH | NOV '83 |
| CONCENTR | GAME | MAR '82 | HIDMAZE | GAME | DEC '82 | POLYCOPY | COPY PROGRAM | NOV '83 |
| SHOOTSTR | GAME | APR '82 | THUNBIRD | GAME | JAN '83 | STOCKMKT | BUSINESS/FIN. | NOV '83 |
| SPACEFRT | GAME | APR '82 | AUTOMATE | AUTOLOAD PGRM. | JAN '83 | NITFLYER | GAME | DEC '83 |
| INCOMEPR | BUSINESS/FIN. | JUN '82 | JUGGLER | GAME | JAN '83 | CHARTER | GRAPH PRGM. | DEC '83 |
| OUTPOST | GAME | JUN '82 | COPYCAT | GAME | FEB '83 | MLX | USR LOAD PRGM. | DEC '83 |
| MAZERACE | GAME | JUL '82 | HORSERAC | GAME | FEB '83 | MOZART | MUSIC PGRM. | JAN '84 |
| MICRODOS | MINIDOS | JUL '82 | SKI | GAME | FEB '83 | COLORBOT | GAME | JAN '84 |
| GOLDRUSH | GAME | JUL '82 | SCRIPTOR | WORD PROCESSOR | APR '83 | | | |
| LASERBAR | GAME | OCT '82 | VIDEO 80 | 80 COLUMN | APR '83 | | | |